Flow Control for Multiple-Access Queues

R. Cristescu and S. D. Servetto[1]
Laboratoire de Communications Audiovisuelles
Ecole Polytechnique Fédérale de Lausanne
CH-1015 Lausanne, Switzerland.
URL: http://lcavwww.epfl.ch/ccn/

Abstract — **We study the problem of finding a characterization for the channel that results when a queue is operated under multiple access conditions. In such systems, the mechanism by which different sources gain access to the channel plays a fundamental role in defining what is the channel available to each source. In this paper therefore we study the structure and properties of these control devices in some detail. Under some (mild) technical conditions, and under modeling assumptions inspired by TCP/IP's flow control (the standard control algorithm in the current Internet), we are able to characterize the optimal controller for this problem. We also present some numerical simulations, to help develop an intuition on what exactly this control box does.**

I. Introduction

A *Bits Through* Multiple Access *Queues*

Let $S^{(1)}...S^{(N)}$ denote N information sources. Each $S^{(i)}$ can be in one of two possible states: when in the ON state, symbols (drawn from a finite alphabet) are generated according to some unspecified distribution; when in the OFF state, no symbols at all are generated. Source state transitions between ON/OFF states are independent over time. The symbols generated by each $S^{(i)}$ are placed in the buffer of a single-server queue, which serves them using some predetermined scheduling algorithm (typically, first-in first-out). In this proposed scenario, two most important questions arise:

- *What is a fair split of the service rate of the queue among the different sources?* Note that this is essentially equivalent to the classical problem of *flow control* in networks. In a network with several users and interconnecting nodes, the need for flow control –i.e., for controlling the amount of data that each source is allowed to inject into the network– arises because of the limited available network resources. The purpose of flow control is to allocate these resources efficiently, while keeping this allocation *fair* among users. Classical papers on this subject are [7, 9], among others.
- *What is the Shannon capacity of the channel available to each source?* In [2], Ananthram and Verdú answer this question in a special case of the above described setup: a single source (i.e., $N = 1$), always in an ON state.

In this work we formulate and present the first steps towards solving a multiuser version of the Bits Through Queues problem of [2], which we refer to as the *Bits Through Multiple-Access Queues* problem. Note that although certainly related, our problem is significantly more complex than that considered in [2]. To start with, under the assumptions of [2] the first question is meaningless, since there is only one source; yet in the general case, the mechanism used to split the queue resources among information sources will essentially determine the channel available to each one of them, and hence its capacity. Further complications arise from the fact that, in the general case, the number of active sources changes over time. A study of flow control techniques suitable for use in the context of our multiuser version of the problem considered in [2] is the main focus of this paper.

B *Distributed Flow Control with Partial Information*

In the design of the desired control modules, there is a wide range of options in terms of information available to the controller, to the sources, to the queue itself, etc. Two extreme examples correspond to cases when (a) there is a unique, global, central controller which can observe exactly the state of the queue and of all the sources at any point in time; and (b) a decentralized, local controller which can only see the state of the individual source it controls, as well as some feedback information that he obtains about the state of the queue. This situation is illustrated in Fig. 1.

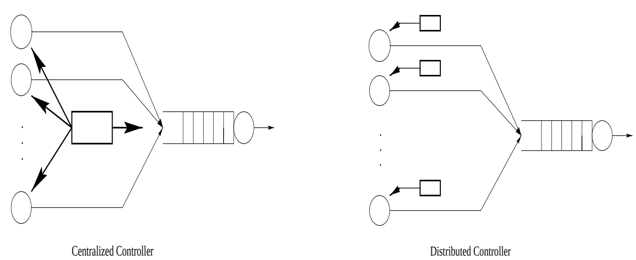

Figure 1: To illustrate two extreme cases in which control may be necessary. Left: a unique centralized global controller; right: local decentralized controllers. In general, global controllers are desirable since they will give the best network performance. The problem is that their communication complexity often renders them prohibitively complex in practice, thus the interest in local decentralized control devices.

Observe in Fig. 1 that, for the centralized controller, the solution to any reasonable formulation of the flow control problem is indeed trivial: knowing all the information about the state of individual sources and of the buffer state, it is enough to allocate to each user a service share equal to the ratio between the service rate and the number of sources active at a given moment. The second type of controller, more realistic but harder to implement, is what we focus on in this work.

[1]Work supported in part by the Swiss National Science Foundation, under grant 21-61831.00.

to yield to mathematical analysis in the context of the *Bits Through Multiple-Access Queues* problem, yet also rich enough to be a good model for real-life situations.

C Main Contributions and Organization of the Paper

Two important contributions are presented in this paper:

- The definition itself of the *Bits Through Multiple-Access Queues Problem.* The model we set up is a good approximation for many network problems, in which many sources have to share a common router. Plus, this is done under assumptions which are an abstraction of situations typically encountered in real-life networks (like the Internet).
- A characterization of the optimal control strategy that multiple sources competing for access to a shared queue should use to access the queue.

We feel what makes our problem formulation interesting in itself –and different from the one considered in [2]– is precisely the shared, multiple access nature of our channel. To the best of our knowledge, problems of sharing network resources (such as flow control [9]) had not been considered before in an information theoretic setting.[1]

The rest of this paper is organized as follows. In Section II we present a dynamical system model for the network. In Section III we give a formal statement for the optimal control problem, and we develop solutions in two cases: (a) assuming a fully observed state, and (b) in the presence of partial information only. In Section IV we present a number of simulation results, where the performance of the proposed controller was tested with numerical examples. Finally, in Section V, final remarks are presented, and future work is discussed.

II. System Model

A Intuition

Consider the following *discrete-time* model:[2]

- At time k, source $S^{(i)}$ generates a symbol with probability $u_k^{(i)}$, and remains silent with probability $1-u_k^{(i)}$. The control task consists of choosing values for all $u^{(i)}$'s, at all times.
- Switching between ON/OFF states for any source occurs independently from the states of other sources.
- The service rate of the queue is deterministic.
- The queue has a finite buffer. When a source generates a symbol to put in this buffer, if the buffer is full then the symbol is dropped and the source is notified of this event; if there is room left in the buffer the symbol is accepted, and the source is notified of this event as well.[3]

[1]For example, a recent survey has pointed out precisely how little impact information theory has had in the networking community [6].

[2]Although the original problem calls for a *continuous-time* formulation, many challenging problems arise in discrete-time as well, and so we decided to start with the latter as a means of managing the complexity of the task at hand.

[3]Note that we could have assumed the buffer size to be infinite, with notification when the occupancy level of this buffer exceeds a fixed threshold value. From the point of view of the design of a controller, both formulations seem equivalent.

to coordinate their efforts in order to choose an appropriate set of control actions $u^{(i)}$ ($i = 1...N$): instead, the only cooperation we allow is in the form of having all sources implement the same control technique, based on partial information about the state of the queue that they have access to. This approach is inspired by the mechanics of TCP's flow control [9], and is a major difference we have found with some of the previous theoretical work on flow control (e.g., [1, 4, 11, 12, 13]). An illustration of the proposed model is shown in Fig. 2.

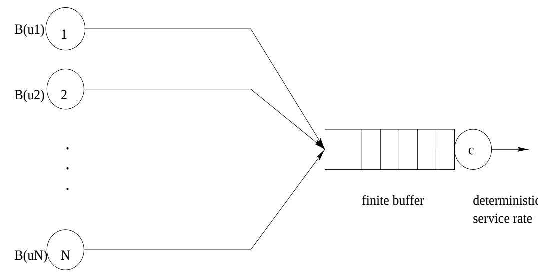

Figure 2: To illustrate the proposed model. N sources switch between ON/OFF states, and generate symbols with a (controllable) probability $u_k^{(i)}$. The only information a source has about the network is a sequence of 3-valued observations: acknowledgments, if the symbol was accepted by the buffer, losses if it is rejected due to overflow, and nothing if the decision was not to transmit at the current moment (denoted 1, -1, 0, respectively).

A fundamental observation, playing an enabling role in the analysis presented later in this paper, is that due to the independence of ON/OFF switching times among sources, *the variation in time of the number of active sources is a Markov process.* Intuitively, this is because the number of active sources at a given time depends only on the number of active sources at the previous time, and on the difference between the number of sources that start/cease transmission at the current time. Denoting by x_k the number of active sources at moment k, we may write

$$x_{k+1} = x_k + t_k - c_k \stackrel{\Delta}{=} x_k + e_k,$$

where t_k, c_k are the number of sources starting/ceasing transmission at time k. Now, since t_k and c_k are iid sequences, so is $e_k \stackrel{\Delta}{=} t_k - c_k$. But sums of iid sequences are Markov sequences, qed [3].

B System Dynamics

Our system is described at time k by the following parameters:

- $x_k \in \{1, \ldots, N\}$ - *hidden* chain states - number of flows *active* at time k.[4]
- $r_k^{(i)} \in \{-1, 0, 1\}$ observations (as defined above); i are the indexes of the sources active at moment k.
- $u_k^{(i)} \in (0, 1]$ source intensities, controllable (as defined above).

[4]Note that we do not include in our system model the case of *zero* active sources, since in that case there is nothing to control!

- Transition probabilities among the hidden states $p(x_k = v|x_{k-1} = s)$ (independent of the source intensities $u_k^{(i)}$), forming the transition matrix $[P]_{sv}$
- $\Pr(x_0)$ - the initial probability density over the states.

- Information available via measurements:
  - $p(r|x, u)$: the probability of occurrence of an observation $r \in \{-1, 0, 1\}$, when x sources are active, and when symbols are generated at rate u.

Note that in setting up this model we have made somewhat of a simplifying assumption. Suppose at time k there are $x_k = s$ sources active, then some of them start/cease transmission, and at time $k + 1$ we have $v \neq s$ active sources. Will $p(r|v, u)$ at moment $k + 1$ be an accurate description of the probabilities with which observations will occur? Not necessarily, since it will take some time until all the remaining active sources adjust their injection rates accordingly. However, after that transient period and before the next state transition, $p(r|v, u)$ will be an accurate description again. The simplifying assumption to which we make reference above consists of neglecting these transients, and take them as model inaccuracies: for as long as the chain does not switch states wildly, this assumption should not pose problems. This issue is explored extensively via numerical simulations (see Section IV and [5]). The model itself is illustrated in Fig. 3.

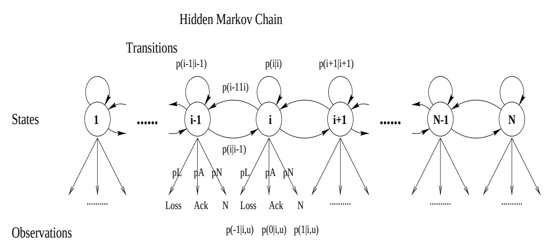

Figure 3: An illustration of the model from the point of view of a single source, based on a simple birth-and-death chain for the evolution of the number of active sources.

Based on the assumptions stated above, the dynamics of the system are given by:

$$
\begin{aligned}
x_{k+1} &= f(x_k, e_k) \qquad (1)\\
r_k^{(i)} &= g(x_k, u_k^{(i)}, e_k^{\prime(i)}), \qquad (2)
\end{aligned}
$$

for $k \geq 0$ and i the indexes of the active sources. Here, f and g are the state transition and observation functions of the (hidden) Markov chain, e_k, e'_k are iid sequences that drive this state-space model. In what follows, we will neglect index i, considering the system from the point of view of a particular active source, the others employing similar behaving individual controllers. A typical sequence of events is illustrated in Fig. 4.

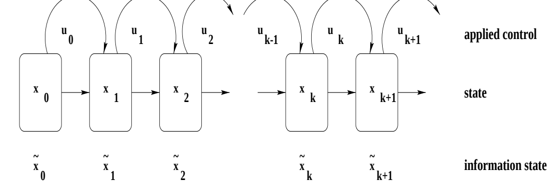

Figure 4: System dynamics. At time k the system is in state x_k. Then the value of a control action u_k is chosen, given the information available up to that point (i.e., the sequence of past observations $r_0, \ldots, r_{k-1}$). Then a new observation r_k is generated, and a state transition occurs.

III. The Control Problem

A Formal Problem Statement

Consider a transmission that is going to last for K time units. Our goal will be to maximize the average throughput, subject to a constraint on the probability of losing symbols. Formally, this is stated by saying that we seek to find a policy $g = \{u_0, \ldots, u_{K-1}\}$ that solves

$$\max_g \frac{1}{K} \sum_{k=0}^{K-1} p(r_k = 1|x_k, u_k) \qquad (3)$$

under the constraints

$$p(r_k = -1|x_k, u_k) \leq T \qquad (k = 0, \ldots, K-1), \qquad (4)$$

for a fixed threshold parameter $T \in (0, 1]$.

In the remainder of this section two solutions to this problem will be presented. In the first one, the state x_k will be assumed to be known. Our interest in this case is because although this is not a good model for our system, the construction of the controller for this system naturally precedes the construction of our sought controller. Finally, we will develop the case of interest.

B Solution under the Assumption of a Fully Observed State

Suppose first that the state x_k is known at moment k, and keep in mind that the state x_k is independent of the control sequence $u_k, k = 0, \ldots, K-1$. Denote the N-dimensional column vector $c(u) = [c(1, u), \ldots, c(N, u)]^T$ (with $c(x, u) = p(1|x, u)$), and an (arbitrary) terminal utility $c_K = [c_K(1), \ldots, c_K(N)]^T$. We then define the "cost-to-go" function V by

$$
\begin{aligned}
V_K &= c_N \qquad (5)\\
V_k &= \sup_u \{c(u) + P(u)V_{k+1}\} \qquad (6)
\end{aligned}
$$

where V_k, V_K are N-dimensional column vectors, the domain of optimization over u is derived from the constraints (4), and the supremum in u is taken separately for each component of the vector equation. It is well known that an optimal policy is given by a sequence u_k that achieves the maximum for each $i = x_k$ [10]. But in the specific case of our system, state transitions do not depend on the control, and so the transition matrix is just $P(u) = P$. That means the optimum is achieved for all V_k for $\sup_{u \in U}\{c(u)\}$ with U determined by the constraints (4). So we see that the cost (3) is optimized when greedy, step-by-step maximization is carried out.

$p(r=1|x,u)$ are monotonically non-decreasing functions of u. Hence, (3) is maximized when the constraints (4) become equalities. If any u_k satisfies (4) with inequality, then we can choose a bigger u_k until equality is reached, and then the corresponding term in the overall cost function will also increase. Therefore, the optimal policy $g=\{u_0,\ldots,u_{K-1}\}$ is given by any solution of

$$p(r_k=-1|x_k,u_k)=T. \tag{7}$$

C Solution with Partial Information

This case is slightly more complicated, since now we assume the state sequence x_k cannot be observed. The problem in this case is that Markovian control policies based on state estimates are not necessarily optimal. However, we do know that optimal policies depend only on an *information state* $\tilde{x}$, which satisfies:

- $\tilde{x}_k$ is a function of $r_0,r_1,\ldots,r_{k-1},u_0,u_1,\ldots,u_{k-1}$.
- $\tilde{x}_{k+1}$ can be determined from $\tilde{x}_k,r_k,u_k$.

Essentially, $\tilde{x}_k$ should contain all the information about x_k that can be inferred from the observations, and it should be possible to compute it by updating the previous information state with the latest available data. Such policies (depending on $\tilde{x}_k$ only) are called *separated* policies, for obvious reasons. A typical choice is to let $\tilde{x}_k$ be $\Pr(x_k|r^{k-1},u^{k-1})$, the conditional probability of x_k given all the past observations and controls applied [10].[5]

C.1 Optimization Function and Constraints

We define the new utility function to be $c(\tilde{x},u)=p(r=1|\tilde{x},u)$, which, using the total probability law, can be expanded as

$$\begin{aligned}c(\tilde{x}_k,u_k) &= p(r_k=1|\Pr(x_k|r^{k-1},u^{k-1}),u_k)\\ &= \sum_{x_k} p(r_k=1|x_k,\Pr(x_k|r^{k-1},u^{k-1}),u_k)\\ &\quad p(x_k|\Pr(x_k|r^{k-1},u^{k-1}),u_k)\\ &= \sum_{x_k} p(r_k=1|x_k,u_k)p(x_k|\Pr(x_k|r^{k-1},u^{k-1}),u_k).\end{aligned}$$

Next we note that, since u_k does not affect x_k and only influences r_k and later observations,

$$p(x_k|\Pr(x_k|r^{k-1},u^{k-1}),u_k)=p(x_k|r^{k-1},u^{k-1}), \tag{8}$$

so finally we have

$$\begin{aligned}&c(\tilde{x}_k,u_k)\\ &\quad= \sum_{x_k} p(r_k=1|x_k,u_k)p(x_k|r^{k-1},u^{k-1})\\ &\quad\triangleq E_{\tilde{x}_k}c(x_k,u_k). \end{aligned}\tag{9}$$

This new utility function has a nice interpretation: it is the average of the functional for the fully observed case at step k, where the average is taken relative to the measure on states given by $\tilde{x}_k$. Similarly, the new constraints are

$$E_{\tilde{x}_k}p(r_k=-1|x_k,u_k)\le T \qquad (k=0,\ldots,K-1). \tag{10}$$

[5] Note: the derivation of the recurrence equations for $\tilde{x}$ is omitted due to its length, and its rather elementary nature (but it can be found in [5]). Here we only mention that indeed, $\tilde{x}_{k+1}=F[r_k,u_k,\tilde{x}_k]$, where F is a linear function up to a normalization term.

The solution of our new constrained optimization problem satisfies a set of Dynamic Programming (DP) equations [10], in which the constraints come into play by restricting the feasibility of certain policies. A control sequence $g=\{g_0,\ldots,g_{K-1}\}$ is said to be *feasible* if

$$u_k=g_k(r_0,\ldots,r_{k-1})\in U_k \tag{11}$$

where U_k is the control constraint set at time k.

Define, for each probability measure π on N elements, the cost-to-go function V as:

$$\begin{aligned}V_K(\pi) &= E_\pi[c(x_K)] \qquad (12)\\ V_k(\pi) &= \sup_{u\in U_\pi} E_\pi[c(x_k,u)+V_{k+1}(F[r_k,u,\pi])]\\ &= \sup_{u\in U_\pi}[\sum_i c(i,u)\pi(i)+\\ &\quad+\sum_{r_k=-1}^{1} V_{k+1}(F[r_k,u,\pi])p(r_k|\pi,u)], \qquad (13)\end{aligned}$$

where

$$U_\pi=\{u|E_\pi p(-1|x,u)\le T\}. \tag{14}$$

The set of optimization U_π is given by the constraint that has to be imposed at present step and at future steps given the possible evolution of the information state, and it is deduced from inequality (10). Since $p(-1|x,u)$ is a non decreasing function of u, for any x we have that

$$U_\pi=(0,u_{\max_\pi}], \tag{15}$$

where $u_{\max_\pi}$ is the unique solution of equation

$$E_\pi p(-1|x,u)=T. \tag{16}$$

Then a policy is *optimal* if u_k achieves the supremum for $\pi=\tilde{x}_k$.

C.3 Practical Difficulties

There are some practical difficulties to implement and simulate the optimal controller in the partial information case as defined above, having to do with the fact that our state space is the whole simplex of probability distributions over N symbols (N is the number of sources in the system). For simulation of these equations on a computer, that simplex has to be discretized. However, a trivial discretization consisting of performing scalar quantization of each entry in π leads to an exponential number of states. Therefore, appropriate vector quantization techniques are required for this purpose. The development of such a simulation is the current focus of our research work. However, a simple, approximate solution exists, which we use in this paper to report some numerical results on the performance of the proposed controllers, inspired by the case of complete information.

The approximation is based on a simple observation. If a control u_k is chosen at time k, it does not influence the next state x_{k+1}, which changes independently. Instead, it will influence the value of the observation r_k, which in turn will determine the update to the information state $\tilde{x}_{k+1}$. Hence, our approximation consists of choosing the maximum control at moment k that still obeys the loss constraint, since this will maximize the throughput and the probability of getting information about the state of the network. Thus we perform solving for u_k

$$E_{\tilde{x}_k} p(r_k = -1 | x_k, u_k) = T. \quad (17)$$

Note that $\tilde{x}_k$ is computed at step k with the information available up to that moment.

Although we do not have a formal proof yet, because control only influences memoryless observations, we suspect that the greedy controller will be optimal in the partial information case as well. This subject is further discussed in [5].

IV. Numerical Simulations

In this section we present results obtained in numerical simulations of the proposed controller. We consider a birth-and-death transition probability matrix P with the parameter p being the probability of the event that the number of active sources increments or decrements. We further assume that p does not depend on the number of sources active at a given moment.

In a first experiment, we study the influence of the different observations on the next information state. We find that the effect of a positive acknowledgment ($r = 1$) is to shift the probability mass in $\tilde{x}$ towards the region of a small number of active sources and the next control will be bigger than the previous one. A negative ack ($r = -1$) though has the opposite effect (mass moves toward states with many sources, and next control will be smaller). In the case when nothing is sent ($r = 0$), the probability density remains basically the same (except for the effect of the transition probability matrix P). This oscillatory effect is intuitively very pleasing, and it resembles the dynamics of TCP. The effect is illustrated in Fig. 5, where different updates over time of the information state are shown, from the point of view of a fixed source.

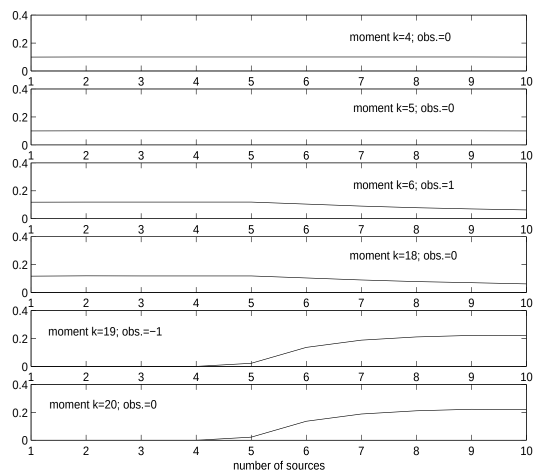

Figure 5: Probability over the number of active sources for a fixed source, at different times, for parameters: $T = 0.04$, $p = 0.001$, $N = 10$ (p is the probability that the chain switches to one more or one less active source, in a birth-and-death model). Observe the shifts in probability mass: positive ack at time 6, negative ack at time 19.

In a second experiment, we want to show the different controller behaviors as a function of the loss parameter T. We find that for high thresholds the controller is more 'optimistic', being able to adapt fast to changes in the number of active rates. Alternatively, when the thresholds are low, inefficiencies may occur in the form of underutilizing the network (when a few sources cease transmission, but the controller takes a long time to adapt to these new conditions). In Fig. 6 these opposite effects are illustrated, with experiments for large and small values of the threshold.

We see that when the probability of state switching is relatively high, sources start or cease transmission frequently, and the controller cannot follow closely the changes in the environment conditions, because it has to estimate the state probability density over a small time interval. The third experiment puts in evidence the capacity of adaptation of local controllers when the transition probability is relatively small, even for a medium valued threshold. The bottom subfigure in Fig. 6 illustrates how the state change is more easily followed in this case by our fixed source.

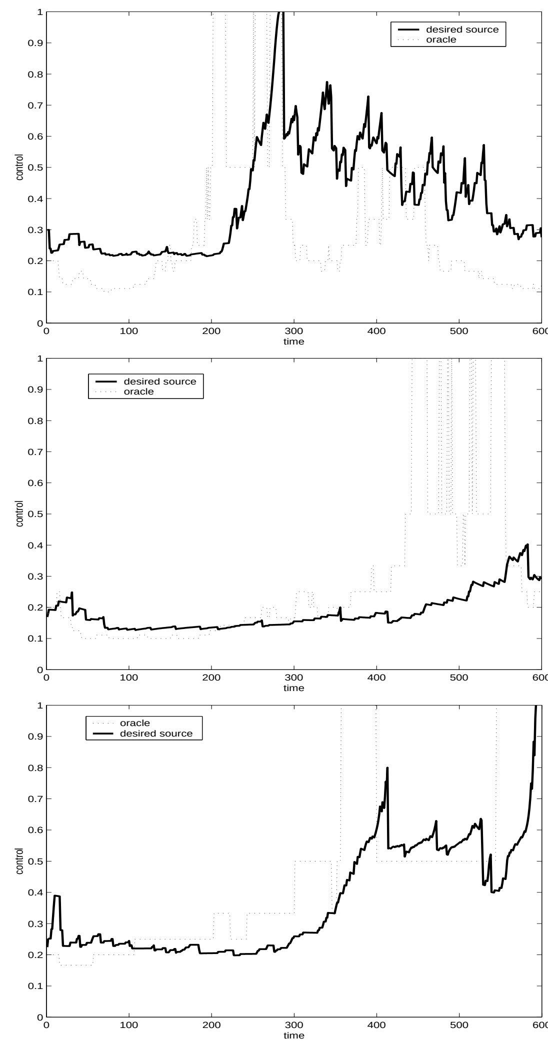

Figure 6: Control sequences for a fixed source and two different thresholds, a symmetric birth-and-death chain with transition probability p and $N = 10$ sources. Up: $T = 0.1, p = 0.1$, middle: $T = 0.02, p = 0.1$, down: $T = 0.05, p = 0.02$. "Oracle" refers to an ideal controller which can actually observe the hidden state.

observe now all the active sources. Fig. 7 plots the maximum and the minimum controls observed at a given moment, no matter which sources exhibit them. It can be seen that while not all the sources transmit at the bandwidth fair share at a certain time step, the difference among the controls stays small and their values oscillate around the fair share, while no source remains privileged over a long period of time.

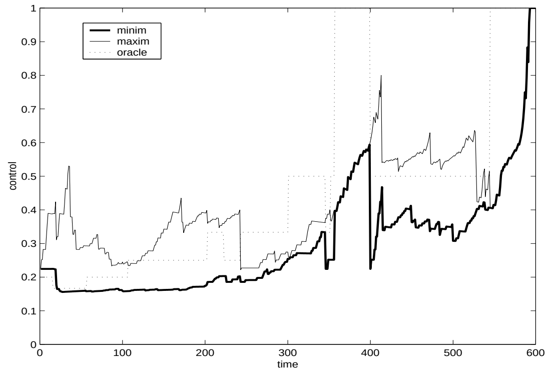

Figure 7: Minimum and maximum values of the control set over time. Note: the sources transmitting at minimum/maximum value are not necessarily the same at different time moments.

One last example of a simple network with $N = 2$ sources shows how the resulting controllers can simultaneously be fair with respect to the other active sources and adapt reasonably to changing activity levels. Fig. 8 shows that when the state does not change very fast, a good choice of the threshold may lead to good performance results.

V. Conclusions

In this work we have shown the structure of an optimal controller to be used by multiple sources to gain access to a shared queue, under the assumption that the sources are independent, switch between ON/OFF states over time, and are not allowed to communicate with each other. We regard this as the first step in coming up with a complete characterization of this multiple-access channel, one that can be used to study capacity and coding problems in this context, along the lines of [8].

Acknowledgments

The authors wish to thank David Tse, for a most useful (and thought provoking) discussion at an early stage in the development of this work.

References

[1] E. Altman. Flow Control Using the Theory of Zero Sum Markov Games. *IEEE Trans. Autom. Control*, 39(4):814–818, 1994.

[2] V. Ananthram and S. Verdú. Bits Through Queues. *IEEE Trans. Inform. Theory*, 42(1):4–18, 1996.

[3] P. Bremaud. *Markov Chains*. Springer-Verlag, 1999.

[4] B. Chakravorti. Optimal Flow Control of an M/M/1 Queue with a Balanced Budget. *IEEE Trans. Autom. Control*, 39(9):1918–1921, 1994.

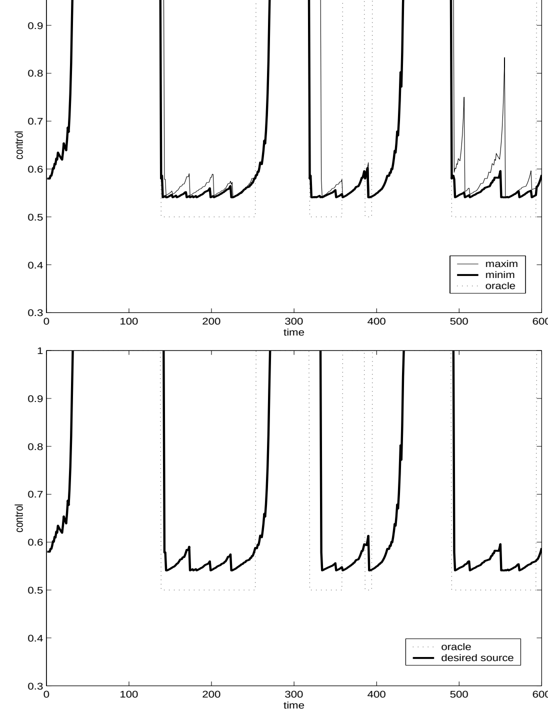

Figure 8: Control sequences for $p = 0.02$, $N = 2$ sources, and threshold $T = 0.04$. Up: minimum/maximum values, down: values for a fixed source.

[5] R. Cristescu and S. D. Servetto. Distributed Flow Control with Partial Information. Technical report, Dept. Systèmes de Communication, Ecole Polytechnique Fédérale de Lausanne, November 2000.

[6] A. Ephremides and B. Hajek. Information Theory and Communication Networks: An Unconsummated Union. *IEEE Trans. Inform. Theory*, 44(6):2416–2434, 1998.

[7] S. Floyd and V. Jacobson. Random Early Detection for Congestion Avoidance. *IEEE/ACM Trans. Networking*, 1(4):397–413, 1993.

[8] A. J. Goldsmith and P. Varaiya. Capacity, Mutual Information, and Coding for Finite-State Markov Channels. *IEEE Trans. Inform. Theory*, 42(3):868–886, 1996.

[9] V. Jacobson. Congestion Avoidance and Control. In *Proc. ACM SIGCOMM*, 1988.

[10] P. R. Kumar and P. Varaiya. *Stochastic Systems: Estimation, Identification and Adaptive Control.* Prentice Hall, Inc., 1986.

[11] R. La and V. Ananthram. Network Pricing Using Game Theoretic Approach. In *Proc. 38th Conf. Dec. Contr.*, 1999.

[12] J. Lei and U.Ozguner. Combined Decentralized Multi-Destination Dynamic Routing and Real-Time Traffic Light Control for Congested Traffic Networks. In *Proc. 38th Conf. Dec. Contr.*, 1999.

[13] Z. Wang, F. Khorrami, and Z.-P. Jiang. Decentralized Robust Control for a Class of Large-Scale Nonlinear Systems. In *Proc. 38th Conf. Dec. Contr.*, 1999.